

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-247466

(43)Date of publication of application : 19.09.1997

(51)Int.Cl.

H04N 1/41
G06F 11/10
H03M 7/40
H04N 7/24

(21)Application number : 08-047665

(71)Applicant : CANON INC

(22)Date of filing : 05.03.1996

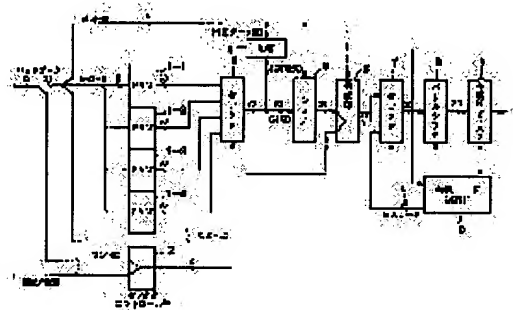
(72)Inventor : MITA YOSHINOBU

(54) ENCODING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To insert a special code in terms of hardware to a bit stream by the addition of a variable length code and the shaving of the same synthesizing circuit, etc.

SOLUTION: An encoding device includes the configuration of a Huffman encoder and a special code inserting circuit. The output of a synthesizing part 6 is normally given to a barrel shifter 8 with a selector 7 but the special code is given from a special code control part 10 to the selector 7, so that one of the output of the synthesizing part 6 or the special code is selected in the selector 7 so as to be given to the barrel shifter 8. The special code is inserted between Huffman code strings by the selection output of the selector 7. In the barrel shifter 8, shift in the proper number of bits is executed as against inputted data in order to generate continuous bit strings in a free area in a synthesizing register part 9 and, after that, an output is executed to the synthesizing register part 9. The synthesizing register 9 is provided with two registers.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

CLAIMS

[Claim(s)]

[Claim 1] The coding equipment carry out having the control means which control in a coding means are coding equipment which performs variable length coding per predetermined pixel block, and perform variable length coding for every predetermined unit within the aforementioned pixel block, a connection means connect the code data generated by the aforementioned coding means, and form a sign bit train, an insertion means insert a predetermined code in the aforementioned sign bit train, and the aforementioned connection means and the aforementioned insertion means as the feature.

[Claim 2] It is coding equipment according to claim 1 characterized by generating the aforementioned predetermined code in the aforementioned control means, and for the aforementioned insertion means choosing the output of the aforementioned coding means, and either of the aforementioned predetermined codes according to control of the aforementioned control means, and outputting to the aforementioned connection means.

[Claim 3] The aforementioned connection means is coding equipment according to claim 2 characterized by having two or more maintenance meanses to hold the inputted data temporarily.

[Claim 4] The aforementioned connection means is coding equipment according to claim 3 characterized by having the two aforementioned maintenance meanses.

[Claim 5] The aforementioned connection means is coding equipment according to claim 4 characterized by having a double buffer.

[Claim 6] Furthermore, it is coding equipment according to claim 4 characterized by inputting the shift output having a shift means to shift the data outputted by the aforementioned insertion means in consideration of the connection situation in the aforementioned connection means, and according [the aforementioned connection means] to the aforementioned shift means.

[Claim 7] The aforementioned shift means is coding equipment according to claim 6 characterized by performing the shift according to the empty situation of the aforementioned maintenance means.

[Claim 8] The aforementioned shift means is coding equipment according to claim 7 characterized by shifting two stages.

[Claim 9] The aforementioned shift means is coding equipment according to claim 8 characterized by dividing a shift amount into a high order bit and a lower bit, and shifting, respectively.

[Claim 10] The aforementioned shift means is coding equipment according to claim 6 to 9 characterized by being a barrel shifter.

[Claim 11] The aforementioned control means are coding equipment according to claim 1 characterized by inserting a predetermined code after this predetermined byte, and connecting again the aforementioned predetermined byte of the aforementioned sign bit train or subsequent ones when a predetermined byte is detected in the sign bit train formed in the aforementioned connection means.

[Claim 12] It is coding equipment according to claim 11 characterized by performing detection of the aforementioned predetermined byte by the AND gate.

[Claim 13] It is coding equipment according to claim 11 characterized by performing detection of the aforementioned predetermined byte for every byte with a register.

[Claim 14] The aforementioned predetermined byte is coding equipment according to claim 11 to 13 characterized by being "FF" byte.

[Claim 15] The aforementioned predetermined code is coding equipment according to claim 1 or 14 characterized by being a code for bit stuffing.

[Claim 16] The aforementioned predetermined code is coding equipment according to claim 1 characterized by being a control code.

[Claim 17] The aforementioned predetermined code is coding equipment according to claim 1 characterized by being a code for a fill bit.

[Claim 18] The aforementioned coding means is coding equipment according to claim 1 characterized by performing Huffman coding.

[Claim 19] It is coding equipment which is equipped with the following, the aforementioned control

means insert the 2nd code after this predetermined byte when a predetermined byte is detected in the sign bit train formed in the aforementioned connection means, and is characterized by connecting again the aforementioned predetermined byte of the aforementioned sign bit train or subsequent ones. A coding means to be coding equipment which performs variable length coding per predetermined pixel block, and to perform variable length coding for every predetermined unit within the aforementioned pixel block. A connection means to connect the code data generated by the aforementioned coding means, and to form a sign bit train. An insertion means to insert the 1st code in the aforementioned sign bit train. Control means which control the aforementioned connection means and the aforementioned insertion means.

[Claim 20] It is coding equipment according to claim 19 which the 1st code of the above is a control code, and is characterized by the 2nd code of the above being a code for bit stuffing or a fill bit.

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] this invention relates to the coding equipment which inserts a special code in the case of variable length coding, concerning coding equipment.

[0002]

[Description of the Prior Art] In order to perform more efficient data communication and more efficient data storage with development of a communication device in recent years or an information processor, coding of data serves as indispensable technology and the various coding methods are proposed.

[0003] The coding method is roughly divided into the fixed-length-coding method whose sign is a fixed length, and the variable-length-coding method whose sign is not necessarily a fixed length.

[0004] In a variable-length-coding method, the method of inserting a special code as a control code into a sign is often used.

[0005] Usually, in case data communication is performed, for example, a noise mixes into a sign, and when the reversal which is a bit occurs, once a sign will be confused, the decode of the sign after it will become impossible. However, according to the coding method which inserts a special code, it becomes possible by discovering the special code in a sign to decode again from the sign after this special code.

[0006] Moreover, the decode method of the sign after this special code can also be changed into a decode equipment side by giving the directions function of decode conditions to a special code.

[0007] Thus, in the variable-length-coding method which inserts a special code, flexible coding is possible.

[0008] In order to insert a special code into the bit stream (bit string) of a variable length sign, a procedure differs from generation of the usual bit stream. For example, once generating a bit stream, the special code was inserted in soft, or coding was stopped in the generation process (coding process) of a bit stream at every special code insertion, and insertion of a special code was realized by the method of inserting a special code with the means like software.

[0009]

[Problem(s) to be Solved by the Invention] However, insertion of the special code by the above-mentioned conventional method had required the remarkable load for CPU which always needs to carry out the monitor of the bit stream, therefore manages coding, in order to insert a special code with the means like software into a bit stream. Moreover, in order to perform insertion like software, the generation rate of a bit stream will be restricted and the limitation had arisen in high-speed processing.

[0010] Of course, although more nearly high-speed coding was attained when insertion of this special code was realizable in hard, in the equipment which performs the conventional coding, there was no concrete means for easy circuitry realizing insertion of a special code in hard.

[0011] Moreover, the circuit scale became large and the barrel shifter used in order to connect a bit string had become the hindrance of hard-izing.

[0012] It is made in order that this invention may solve the technical problem mentioned above, and it aims at offering the coding equipment which enables hard insertion of the special code to a bit stream by small-scale circuitry.

[0013]

[Means for Solving the Problem] The coding equipment of this invention is equipped with the following composition as a way stage for attaining the above-mentioned purpose.

[0014] That is, it is coding equipment which performs variable length coding per predetermined pixel block, and it carries out having the control means which control in a coding means perform variable length coding for every predetermined unit within the aforementioned pixel block, a connection means connects the code data generated by the aforementioned coding means, and form a sign bit train, an insertion means insert a predetermined code in the aforementioned sign bit train, and the aforementioned connection means and the aforementioned insertion means as the feature.

[0015] For example, it is characterized by generating the aforementioned predetermined code in the

aforementioned control means, and for the aforementioned insertion means choosing the output of the aforementioned coding means, and either of the aforementioned predetermined codes according to control of the aforementioned control means, and outputting to the aforementioned connection means.

[0016] For example, the aforementioned connection means is characterized by having two or more maintenance meanses to hold the inputted data temporarily.

[0017] For example, the aforementioned connection means is characterized by having the two aforementioned maintenance meanses.

[0018] For example, the aforementioned connection means is characterized by having a double buffer.

[0019] Furthermore, it has a shift means to shift the data outputted by the aforementioned insertion means in consideration of the connection situation in the aforementioned connection means, and the aforementioned connection means is characterized by inputting the shift output by the aforementioned shift means.

[0020] For example, the aforementioned shift means is characterized by performing the shift according to the empty situation of the aforementioned maintenance means.

[0021] For example, the aforementioned shift means is characterized by shifting two stages.

[0022] For example, the aforementioned shift means is characterized by dividing a shift amount into a high order bit and a lower bit, and shifting, respectively.

[0023] For example, it is characterized by the aforementioned shift means being a barrel shifter.

[0024] For example, when a predetermined byte is detected in the sign bit train formed in the aforementioned connection means, the aforementioned control means insert a predetermined code after this predetermined byte, and are characterized by connecting again the aforementioned predetermined byte of the aforementioned sign bit train or subsequent ones.

[0025] For example, it is characterized by performing detection of the aforementioned predetermined byte by the AND gate.

[0026] For example, it is characterized by performing detection of the aforementioned predetermined byte for every byte with a register.

[0027] For example, the aforementioned predetermined byte is characterized by being "FF" byte.

[0028] For example, it is characterized by the aforementioned predetermined code being a code for bit stuffing.

[0029] For example, it is characterized by the aforementioned predetermined code being a control code.

[0030] For example, it is characterized by the aforementioned predetermined code being a code for a fill bit.

[0031] For example, the aforementioned coding means is characterized by performing Huffman coding.

[0032] Moreover, a coding means to be coding equipment which performs variable length coding per predetermined pixel block, and to perform variable length coding for every predetermined unit within the aforementioned pixel block, A connection means to connect the code data generated by the aforementioned coding means, and to form a sign bit train, It has the control means which control an insertion means to insert the 1st code in the aforementioned sign bit train, and the aforementioned connection means and the aforementioned insertion means. the aforementioned control means When a predetermined byte is detected in the sign bit train formed in the aforementioned connection means, the 2nd code is inserted after this predetermined byte, and it is characterized by connecting again the aforementioned predetermined byte of the aforementioned sign bit train or subsequent ones.

[0033] For example, the 1st code of the above is a control code, and it is characterized by the 2nd code of the above being a code for bit stuffing or a fill bit.

[0034] By the above composition, common composition realized insertion of a special code, and addition of a variable length sign, the load of hardware was mitigated sharply and insertion of a special code was attained at high speed.

[0035] Furthermore, the circuit scale was mitigated by dividing the shift amount in a barrel shifter into two stages.

[0036]

[Embodiments of the Invention] Hereafter, 1 operation gestalt concerning this invention is explained in

detail with reference to a drawing.

[0037] The <1st operation gestalt> The feature of the sign treated in this operation gestalt is explained first.

[0038] The variable length sign treated with this operation gestalt is Huffman coding which constitutes the part in the JPEG standard compression which compresses a multiple-value picture. In coding with this operation gestalt, the operation which inserts an overhead bit between Huffman coding occurs.

[0039] JPEG coding is explained briefly here. First, if it quantizes by carrying out dispersed cosine conversion (DCT conversion) of the multiple-value data for every pixel block, that from which the coefficient after quantization is set to "0" will occur mostly. And a pixel is aligned with a zigzag scan for every block, width of face (this is called category) of coefficients other than zero following it is made a set with 0 run length, and Huffman coding is assigned. And coefficients other than zero (an overhead bit is called) are continued after this Huffman coding. However, in a JPEG sign, Huffman coding and an overhead bit also have the case where do not always appear by turns, for example, overhead bits, such as termination of a block, do not exist.

[0040] Moreover, as a special code which appears in a Huffman code, two or more restart codes etc. exist, for example. When a mistake occurs in a Huffman code in a communication error etc., this restart code is referred to in order to return decoding.

[0041] In order to insert these special codes, a bit "1" is first packed to a byte's break to the bit string of the generated Huffman code. Henceforth, this is called fill byte operation. And after inserting "FF" byte after that, the special code which consists of other than "00" bytes is inserted. That is, "FF" byte is a recognition byte who shows that a special code continues behind.

[0042] Therefore, you have to enable distinction with a special code by inserting "00" bytes intentionally just behind "FF" "FF" byte into the bit stream which consists of a Huffman code and its overhead bit, in the case where "FF" byte occurs by chance, and the case of fill byte operation, when a byte occurs.

[which was generated] Henceforth, this operation is called bit stuffing.

[0043] In the coding equipment of this operation form, it is characterized by making possible fill byte operation mentioned above by easy circuitry, special code insertion, and bit stuffing.

[0044] The block composition of the coding equipment in this operation form is shown in drawing 1 . The coding equipment of this operation form is characterized by including the insertion circuitry of the Huffman encoder and a special code (a marker code is called henceforth).

[0045] In drawing 1 , RAM 1-1 to 1-4 is an encode table for generating a Huffman code (generation), corresponds to compression of the component specified by JPEG standard compression, and consists of four tables for DC component and AC components to the DCT result of a brightness component, a color difference component, and its two components. And a selector 3 is controlled so that the selector control section 2 judges the present processing component and chooses the output of a required table among RAM 1-1 to 1-4.

[0046] The WORD composition of RAM 1-1 to 1-4 which is an encode table is shown in drawing 2 . Since encoding TEPURU of this operation gestalt corresponds to a maximum of 16-bit code, it consists of the 16-bit Huffman-coding field and the field of 4 bits of code lengths, and the Huffman code is stored in the Huffman-coding field by right justification. Addressing to RAM 1-1 to 1-4 at the time of obtaining a Huffman code from these encode tables is performed by the address acquired from the pack data inputted.

[0047] The example of composition of the pack data which are the input of this operation gestalt is shown in drawing 3 here. The pack data of this operation gestalt show the list (the order of a zigzag scan) of DC coefficient and AC coefficient after giving DCT conversion and quantization to a pixel block based on JPEG coding mentioned above. Bit strings, such as DC category in this pack data, AC category, and zero run, may serve as address information to encoding TEPURU as it is. moreover, DC shown in drawing 3 -- difference and AC component are equivalent to the overhead bit mentioned above, and are not used for access of RAM 1-1 to 1-4, but are sent to the synthetic section 6 shown in drawing 1

[0048] As shown in drawing 3 , the discernment flag field which consists of 2 bits exists in pack data, in

the selector control section 2, it is used in the case of distinction of DC component or AC component, and this discernment flag is used for the judgment of whether to be an overhead bit etc. In addition, EOB shown with a discernment flag "11" shows the case where zero run continues to the termination of a block.

[0049] Huffman coding which returned to drawing 1 and was obtained from the selector 3 is shifted leftward (the direction of MSB) by the shifter 5, and overhead bit data are given to the LSB side in the synthetic section 6. The shift amount in a shifter 5 becomes settled by DC category or AC category in pack data. In addition, when an overhead bit does not need to be given to the degree of a Huffman code, the shift amount in a shifter 5 is set to "0."

[0050] Moreover, in an adder unit-4, it is added with the addition data length the code length corresponding to Huffman coding obtained from the selector 3 is indicated to be by DC category or AC category in pack data, and total code length is called for. And this code length is inputted into the special code control section 10.

[0051] By the way, in the synthetic section 6 by which a Huffman code and an overhead bit are compounded, although operation which carries out the mask of the bit of a high order, and is set to "0" from a Huffman code may be performed, even if it does not perform this operation in this operation gestalt, in the register section 9 for composition mentioned later, the bit of a high order is disregarded rather than a Huffman code.

[0052] Although the output of the synthetic section 6 is usually given through a selector 7 to the barrel shifter 8, with this operation gestalt, the special code is given to the selector 7 from the special code control section 10, and it chooses the output of the synthetic section 6, and either of these special codes in a selector 7, and gives them to the barrel shifter 8. The selection output by this selector 7 enables it to insert a special code between Huffman code trains in this operation gestalt.

[0053] In the barrel shifter 8, in order to form the bit string which followed the free area in the register section 9 for composition, after shifting the suitable number of bits to the inputted data, it outputs to the register section 9 for composition. The register section 9 for composition has two registers, when one of the two's register fills, sweeps out the bit string formed in this register, and has the function to hold the input from the barrel shifter 8 to a register with an opening or a register with an opening, and an intact register.

[0054] Hereafter, with reference to drawing 4, operation in the register section 9 for composition is explained still in detail.

[0055] (a) - (c) of drawing 4 is drawing showing typically the structure of the Huffman-coding data which are inputted into a shifter 5 and outputted from the synthetic section 6. First, the Huffman-coding data inputted into a shifter 5 are shown by the slash of (a) of drawing 4, and "0" is put in the portion exceeding the code length of Huffman coding. This "0" may be held at the encoder RAM1-1 grade, and even if it is values other than "0", it does not produce a problem.

[0056] (b) of drawing 4 shows the state where the Huffman-coding data inputted in the shifter 5 were shifted in the direction of MSB by the overhead bit length added to LSB. (c) of drawing 4 shows the state where it was compounded so that it might become the bit string which Huffman-coding data and an overhead bit follow in the synthetic section 6. Here, "0", then the synthetic section 6 can constitute easily the high order bit which puts "0" in the LSB side which was vacant when Huffman-coding data were shifted in the direction of MSB in a shifter 5, and exceeds the bit length of the overhead bit of pack data by the OR gate which takes the OR of two inputs.

[0057] Thus, suppose that a total of b-bit data is made as the synthetic section 6 is shown by (c) of drawing 4. In addition, a selector may be used as the synthetic section 6, and the LSB side may constitute the overhead bit and MSB side so that the output of a shifter 5 may be chosen and a LSB side or the MSB side may be chosen according to overhead bit length.

[0058] (e) of drawing 4 and (f) are drawings showing typically the two contents of a register in the register section 9 for composition, respectively. Here, each register presupposes that it is 32-bit width of face.

[0059] for example, when it obtains as one of the two's register shown in (f) of drawing 4 shows in the

sky (e) of drawing 4 , and the data of a bit length are already stuck for one of the two's register, the shift amount at the time of the Huffman-coding data shown in (c) of drawing 4 in the barrel shifter 8 being shifted serves as $32-(a+b)$ bit. If this shift-amount $32-(a+b)$ becomes negative, the shift direction will mean that the contrary or the shift which added 32 to the negative value is performed.

[0060] In addition, (d) of drawing 4 shows the example of output data after the shift from the barrel shifter 8, and is 32 bits like the data in the register section 9 for composition. In (d) of drawing 4 , when $a+b$ exceeds 32 bits, the example of a shift in case the shift amount in the barrel shifter 8 is negative is shown.

[0061] In the register which shows the output of this barrel shifter 8 to (e) of drawing 4 in the register section 9 for composition, only a $d=32-a$ -bit LSB portion is latched. And in the register shown in (f) of another drawing 4 , a part for a $b-d$ bit is latched to an MSB portion. Thereby, since the register shown in (e) of drawing 4 becomes full, it is processed as data outputted outside from the register section 9 for composition.

[0062] Thus, in the register section 9 for composition, it will output, if one of the two's register fills, and it is made empty. And it becomes possible by repeating the same operation to form the bit string which continued in this operation form. Therefore, it will always have stood by for one of the two's register in the register section 9 for composition by sky condition.

[0063] Here, when $a+b$ does not exceed 32 bits, the example of an output of the barrel shifter 8 in case the shift amount in the barrel shifter 8 is positive is shown in (g) of drawing 4 . Moreover, (h) of drawing 4 shows the state where effective b bits in the output of the barrel shifter 8 were written in one register in the register section 9 for composition. In such a state, the writing to the register of another side of the register section 9 for composition is not performed.

[0064] Moreover, in the special code control section 10, in addition to insertion of a special code, the shift amount of the barrel shifter 8 is judged based on the write-in state of the register section 9 for composition, or control of the output of the continuous bit stream packed from the register section 9 for composition to 32 bits etc. is managed.

[0065] Hereafter, the still more detailed composition of the register 9 for composition explained above is shown and explained to drawing 5 . In addition, in drawing 5 , a part of composition in the special code control section 10 is included, and the boundary of the register 9 for composition and the special code control section 10 is expressed with the dotted line.

[0066] In drawing 5 , a register A21 and a register B22 are equivalent to the 32-bit register of (e) and (f) shown in drawing 4 mentioned above, respectively.

[0067] For example, it supposes that the a -bit code is already written in the register A21, a shift amount $[\text{in} / b \text{ (} b = \text{Huffman-coding length} + \text{overhead bit length)}]$, then the barrel shifter 8 / for the code length of the code written in a degree] serves as $32-(a+b)$ bit, as mentioned above, and a part of output of the barrel shifter 8 is latched to registers A21 and B22.

[0068] more, the mask data to each register is generated by the mask generation sections 23 and 24 constituted in the special code control section 10, and the latch control section 25, and as mentioned above by this mask data, with a register A21, the data except a bits of high orders already written in are latched to a detail -- having -- a register B22 -- $(a+b)-32$ of a high order -- the number of bits is latched for or $(b-d)$

[0069] Here, when $(a+b)-32$ are negative, it is not latched to a register B22. On the other hand, since a register A21 fills with a code at the same time a latch is performed at a register B22 in a positive case, the 32-bit data of a register A21 can begin to come to a latter circuit, and serve as empty. Thereby, a role changes registers A21 and B22.

[0070] Here, mask generation is explained. In the mask generation sections 23 and 24 in the special code control section 10, based on a value a and a value $(a+b)-32$, the write-in position to a register A21 or a register B22 is judged, and a mask data is generated. If the role of a register A21 and a register B22 changes this mask data as mentioned above although it is given through the latch control section 25 to a register A21 and a register B22, exchange will be performed also for two outputs in the latch control section 25.

[0071] AND for every byte is taken in the AND gate 28, and it is judged [32 bits each outputted from registers A21 and B22 i.e., a total of 8 bytes, of / of "FF"], respectively whether you are a byte. In order to except a bits which processing has already ended from a processing object at this time, the mask for a bits is covered by the mask generation section 30 through the gate 34.

[0072] Moreover, since the marker (special code) accompanied by "FF" is also given to registers A21 and B22 through a selector 7 from the special code control section 10 shown in drawing 1 , according to a position, the mask of "FF" in a marker generated by the mask generation section 32 through the gate 34 is covered.

[0073] Furthermore, since there is a possibility that unnecessary information may be stored also about the remaining portion except a bits [finishing / processing] and the bit which is the sum of b bits of Huffman coding under processing (a+b), the mask generated by the mask generation section 31 through the gate 34 is covered.

[0074] Here, although it becomes $anew=a+b$ when "FF" byte is not able to discover, when "FF" byte is discovered, the position f is discovered by the priority encoder 35, and it considers as $anew=f$. That is, even the position of ** "FF" is made finishing [processing].

[0075] The marker which is a special code consists of bytes other than "FF" "0" following a byte and it, and the mask generated by the mask generation section 32 according to the information which the amendment section 33 outputs according to a marker insertion point is covered through the gate 34.

[0076] As explained above, when "FF" byte is discovered, actual processed number-of-bits a is updated, and becomes $anew=f$. Then, in the special code control section 10, "00" bytes' insertion, i.e., bit stuffing, is performed by outputting "00" to a selector 7. And it is necessary to latch again a part for the remaining bit (for you to be c bits) to registers A21 and B22 after that through the barrel shifter 8.

[0077] As shown in drawing 6 , the remaining portion which finished writing in the bit string of a before [bit stuffing] among b bits becomes c bits, and this continues being held here at the input section of the barrel shifter 8. c bits is expressed with the following formulas to b bits of last numbers of processing here.

[0078]

$$c=bnew=b-(f-a) \dots (1)$$

in addition, a formula (1) -- setting (f-a) -- it is the last processed number of bits

[0079] Therefore, in order to process these c bits in the barrel shifter 8, it is $32-(a'+bnew)$ (2)

A ** shift is needed. It is here and is $a'=anew+8$ (3)

Since it is $anew=f$ and the number.of $8=$ SUTAFFINGUBITTO, (2) formulas become $32-(a+b+8)$ from (1) and (3) formulas a' and by substituting bnew to (2) formulas. That is, 8 bits of shift amounts in the barrel shifter 8 will shift for bit stuffing.

[0080] In addition, last processed number-of-bits f-a is obtained from a subtractor 36. And if f usually (a+b) becomes 32 or more at 32 or more and the time of bit stuffing generating, the value (32 bits) of registers A21 and B22 which became full will be outputted through a selector 26, and it will be outputted outside as compressed data through a register 27.

[0081] Moreover, what is necessary is to set difference with the larger multiple of 8 than a and a to b, and just to write in a bit "1" to the register 9 for composition through a selector 7 and the barrel shifter 8 at the time of fill byte operation (just before marker code insertion).

[0082] In addition, distribution of the function of the register 9 for composition and the special code control section 10 does not necessarily need to be as being shown in drawing 5 , for example, may be the composition that AND-gate 28 grade is also contained in the special code control section 10.

Namely, what is necessary is just to be able to fill the composition shown in drawing 5 by the register 9 for composition, and the special code control section 10.

[0083] Next, the detailed composition of the barrel shifter 8 is shown in drawing 7 , and it explains below.

[0084] Although the barrel shifter 8 of this operation gestalt is a 32-bit barrel shifter which has the output of 32-bit width of face, since explanation is simple, the example of composition of a 8-bit barrel shifter is shown in drawing 7 , and it explains to it.

[0085] In drawing 7, 81 is a buffer which receives an input and tells the signal to D0-D7 to the latter part. This signal shifts 1 bit of turn at a time to 8to(es)1 selector 82-1 to 82-8, respectively, and is inputted into it. Therefore, if the value same as a selection-control signal of a selector 82-1 to 82-8 is given, the value which shifted 1 bit at a time will be acquired by S0-S7. Therefore, what is necessary is just to give the shift amount to wish as a selection-control signal over a selector 82-1 to 82-8.

[0086] Therefore, the 32-bit barrel shifter 8 of this operation gestalt should just realize the buffer 81 and selector 82-1 to 82-8 in drawing 7 as 32-bit correspondence.

[0087] By according to this operation gestalt, sharing a circuit for composition, a barrel shifter, etc. which add a variable length sign for the operation relevant to insertion and it of a special code, and realizing, as explained above, the load of hardware was mitigated sharply and high-speed insertion of a special code was attained by small-scale circuitry.

[0088] The 2nd operation gestalt concerning this invention is explained below the <2nd operation gestalt>. In the 2nd operation gestalt, the detailed composition of the register 9 for composition differs from the 1st operation gestalt mentioned above.

[0089] The detailed composition of the register 9 for composition in the 2nd operation gestalt is shown in drawing 8. In drawing 8, the same number is attached about the same composition as drawing 5 shown with the 1st operation gestalt mentioned above, and explanation is omitted.

[0090] In the composition shown in drawing 8, fundamentally, if a register A21 or a register B22 fills, data will be sent to a latter circuit through a selector 26. This structure is as the 1st operation gestalt mentioned above having explained.

[0091] In the 2nd operation gestalt, it has 2 sets of 4-byte registers 41 and 43 as a latter-part circuit of a selector 26, and it writes in by control of the R/W control section 44 and a selector 42, confirming as reading every 1 byte of whether to be "FF" from one of the two's register 41 to the register 43 of another side. The register 43 for writing outputs data to non-illustrated FIFO or the buffer for an output for 4 bytes of every writing.

[0092] As for every 1 byte of data read from the register 41, it is judged through latch 45 at the gate 47 whether it is "FF." If it is "FF", the gate 46 will be closed immediately after writing "FF" in the register 43 for writing, and "00" bytes will be continued and written in the register 43 for writing. Bit stuffing in this, i.e., the 2nd operation gestalt, is performed. However, it is necessary to cancel the result of "FF" discovery, and in the special code control section 10, according to the position of the marker code "FF" which self generated, it is necessary to control so that bit stuffing does not occur. This is attained by the marker generating section 50, and the delay section 49 and the gate 48. In addition, it realizes by the same method as the 1st operation gestalt of "1" by generating of the marker code in the 2nd operation gestalt, and fill byte operation which mentioned insertion above.

[0093] incidentally, if the fill byte.operation in the 2nd operation gestalt sets the processed number of bits in front of marker insertion to a, only the number of bits of $8 - (a \% 8)$ will be attained by giving "1" as an input of the barrel shifter 8 through a selector 7 altogether just because 8 divided a by 8 a% however -- For example, when it is $a = 9$, 7 bits is given to the barrel shifter 8 as "1." Of course, it is also possible to perform fill byte operation and marker code insertion by the barrel shifter 8 simultaneously.

[0094] As explained above, according to the 2nd operation gestalt, the composition of a different synthetic register from the 1st operation gestalt can perform insertion of a special code etc. similarly.

[0095] The 3rd operation gestalt concerning this invention is explained below the <3rd operation gestalt>. In the 3rd operation gestalt, the detailed composition of the register 9 for composition differs from the 1st operation gestalt mentioned above.

[0096] The detailed composition of the register 9 for composition in the 3rd operation gestalt is shown in drawing 9. In drawing 9, the same number is attached about the same composition as drawing 8 shown with the 2nd operation gestalt mentioned above, and explanation is omitted.

[0097] In the 1st operation gestalt mentioned above, when the byte obtained as a result of fill byte operation when a marker code is simultaneously given with a fill byte "1" to the barrel shifter 8 is "FF", you newly have to reset a marker code.

[0098] Moreover, in consideration of the byte set to "FF" as a result of performing a fill byte for marker

code insertion, it is necessary to design carefully with bit stuffing being performed appropriately in the 2nd operation form mentioned above.

[0099] In the 3rd operation gestalt, it is characterized by canceling the starting fault by realizing composition of a marker code by composition which became independent of fill byte operation and bit stuffing.

[0100] Basic operation in the composition shown in drawing 9 is the same as that of the 2nd operation gestalt mentioned above. Therefore, the same is said of insertion operation of the bit stuffing which inserts "00", and the fill byte in front of marker code insertion "1." However, in the 3rd operation gestalt, insertion of a marker code writes 1 byte at a time in the register 43 for writing in order directly through a selector 74 without a selector 7 and the barrel shifter 8. That is, in the 3rd operation gestalt which realizes insertion of a marker code by composition which became independent of fill byte operation and bit stuffing, it is necessary to adjust the timing of a marker code so that a marker code may enter after the byte boundary which inserted a fill byte "1." This adjustment realizes timing in the marker generating section 77 by doubling suitably by the delay section 76 for the fill byte Management Department 75 and timing adjustment. Thereby, it can set in the 3rd operation gestalt and the load to the barrel shifter 8 at the time of marker code insertion can be mitigated.

[0101] Moreover, although it is necessary to detect "FF" byte in the case of the bit stuffing which inserts "00", since it is not necessary to take the position of a marker code into consideration in this case, it judges that it is "FF" to the data which pass the byte register 72 at the gate 71. And if it is "FF", a register 72 will be cleared by one clock and "00" bytes will be inserted. In addition, a register 73 is used as beginning registers, such as a marker code.

[0102] Marker code insertion and bit stuffing can be performed without taking into consideration especially "FF" byte that arose by the fill byte by realizing composition of a marker code by composition which became independent of fill byte operation and bit stuffing according to the 3rd operation gestalt, as explained above.

[0103] The 4th operation gestalt concerning this invention is explained below the <4th operation gestalt>. In the 4th operation gestalt, the example which realizes the 32-bit barrel shifter 8 with the composition of a smaller scale is explained. The detailed composition of the barrel shifter 8 in the 4th operation gestalt is shown in drawing 10.

[0104] First, a buffer 83-1 to 83-4 is a 8-bit buffer, and receives an every 8 bits input, respectively. 84-1 to 84-4 is a selector, four bytes is shifted without changing the turn in 8 bits (1 byte), and 32 bits of turn inputted into this selector, respectively are shifted. That is, rotation in a 8-bit unit is performed.

[0105] Hereafter, operation in a selector 84-1 to 84-4 is explained. A selector 84-1 to 84-4 consists of eight 4to(es)1 selectors, respectively. Four are inputted as a signal (for example, D0, D8, D16, D24) with which 8 bits of signals held at the buffer 83-1 to 83-4, respectively shifted to four to(es)each1 selector, and one of them is chosen. And based on the same selection signal, 1 byte of a 8-bit continuous signal (for example, D0-D8), i.e., signal, is acquired by inputting the signal which shifted 1 bit at a time from other 4to(es)1 selectors, respectively into eight 4to(es)1 selectors which constitute each selector.

[0106] Moreover, a shift in a byte unit will be given to t0-t31 which are outputted by inputting into each selector 84-1 to 84-4 the signal which it shifted 8 bits at a time mutually.

[0107] When the shift amount in the barrel shifter 8 is made into 5 bits (shift amount of 0-31), it means that the shift of 2 bits (8 integral multiples) of shift processings of the byte unit in a selector 84-1 to 84-4, i.e., a high order, was performed first here. Next, although a low rank triplet is shifted (shift amount of 0-7), for that, the data over two bytes are needed. Therefore, in order to take out 8 bits from 16 bits inputted according to a shift amount, four [85-1 to 85-4], i.e., a barrel shifter, are needed for the 16-bit barrel shifter which has eight 16to1 selectors. In addition, if there are four shifters which have eight 8to(es)1 selectors in fact since shift amounts are 0-7 in this case, it will end.

[0108] The signal which shifted 1 bit at a time to each input of 8to(es)1 selector is inputted by 8 bits (for example, t0, t1, ... t8), and the signal with which 1 bit of eight 8to(es)1 selectors shifted from other 8to(es)1 selectors, respectively is inputted. Therefore, 16-bit width of face is required for the input of the barrel shifter 85-1 to 85-4. This is realizable with the same composition as the 8-bit barrel shifter shown

in drawing 7 in the 1st operation gestalt mentioned above, if the difference in an input signal is removed.

[0109] And 8 bits which continued, respectively are further obtained as an output of the 32-bit barrel shifter 8 as a result continuously with the outputs of four barrel shifters 85-1 to 85-4 by giving the value which it shifted 8 bits at a time, respectively to the barrel shifter 85-1 to 85-4. It means that the shift for a total of 5 bits was given in this, i.e., the 4th operation gestalt.

[0110] By making a barrel shifter two-step composition, as are explained above, and shown in drawing 10 according to the 4th operation gestalt, and dividing a shift amount into a high order bit and a lower bit, rather than the case where the composition of a 8-bit barrel shifter shown in drawing 7 in the 1st operation gestalt is extended as a 32-bit barrel shifter as it was, a circuit scale is small and ends.

[0111] Even if it applies this invention to the system which consists of two or more devices (for example, a host computer, an interface device, a reader, a printer, etc.) which are operation gestalt > besides <, you may apply it to the equipments (for example, a copying machine, facsimile apparatus, etc.) which consist of one device.

[0112]

[Effect of the Invention] As explained above, the load to hardware is sharply mitigated by having realized operation relevant to insertion and it of a special code by sharing the same circuit for composition as addition of a variable length sign etc. according to this invention, and the high-speed coding processing by small-scale circuitry is attained.

[0113] Moreover, reduction of the further circuit scale was attained by making a barrel shifter two-step composition and shifting by dividing a shift amount into a high order bit and a lower bit.

[0114]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the composition of the coding equipment in 1 operation gestalt concerning this invention.

[Drawing 2] It is the WORD block diagram of the encoding RAM in this operation gestalt.

[Drawing 3] It is drawing showing the example of pack data in this operation gestalt.

[Drawing 4] It is drawing for explaining operation of the register for composition in this operation gestalt.

[Drawing 5] It is the block diagram showing the detailed composition of the register for composition in this operation gestalt.

[Drawing 6] It is drawing for explaining operation of the register for composition in this operation gestalt.

[Drawing 7] It is drawing showing the detailed composition of the barrel shifter in this operation gestalt.

[Drawing 8] It is the block diagram showing the detailed composition of the register for composition in the 2nd operation gestalt concerning this invention.

[Drawing 9] It is the block diagram showing the detailed composition of the register for composition in the 3rd operation gestalt concerning this invention.

[Drawing 10] It is the block diagram showing the detailed composition of the barrel shifter in the 4th operation gestalt concerning this invention.

[Description of Notations]

1-1 to 1-4 Encoding RAM

2 Selector Control Section

3 Selector

4 Adder

5 Shifter

6 Synthetic Section

7 Selector

8 Barrel Shifter

9 Register Section for Composition

10 Special Code Control Section

21 22 Register

23 24 Mask generation section

25 Latch Control Section

26 Selector

27 Register

28 AND Gate

29 Selector

30, 31, 32 Mask generation section

33 Amendment Section

34 Gate

35 Priority Encoder

36 Subtractor

41 43 Register

42 Selector

44 R/W Control Section

45 Latch

46 Gate

47 AND Gate

48 Gate
49 Delay Section
50 Marker Generating Section
71 AND Circuit
72 73 Register
74 Selector
75 Fill Byte Management Department
76 Delay Section
77 Marker Generating Section
81 Input Section
82-1 to 82-8 Selector
83 Input Section
84-1 to 84-4 Selector
85-1 to 85-4 Selector

[Translation done.]

【特許請求の範囲】

【請求項 1】 所定の画素ブロック単位に可変長符号化を行う符号化装置であって、

前記画素ブロック内の所定単位毎に可変長符号化を行う符号化手段と、

前記符号化手段によって生成された符号データを連結して符号ビット列を形成する連結手段と、

前記符号ビット列に所定コードを挿入する挿入手段と、
前記連結手段と前記挿入手段とを制御する制御手段とを有することを特徴とする符号化装置。

【請求項 2】 前記所定コードは前記制御手段において生成され、

前記挿入手段は、前記制御手段の制御に応じて前記符号化手段の出力と前記所定コードのいずれかを選択して前記連結手段に出力することを特徴とする請求項 1 記載の符号化装置。

【請求項 3】 前記連結手段は、入力されたデータを一時保持する保持手段を複数備えることを特徴とする請求項 2 記載の符号化装置。

【請求項 4】 前記連結手段は、前記保持手段を 2 つ有することを特徴とする請求項 3 記載の符号化装置。

【請求項 5】 前記連結手段は、ダブルバッファを有することを特徴とする請求項 4 記載の符号化装置。

【請求項 6】 更に、前記挿入手段により出力されたデータを前記連結手段における連結状況を考慮してシフトするシフト手段を有し、

前記連結手段は、前記シフト手段によるシフト出力を入力することを特徴とする請求項 4 記載の符号化装置。

【請求項 7】 前記シフト手段は、前記保持手段の空き状況に応じたシフトを行うことを特徴とする請求項 6 記載の符号化装置。

【請求項 8】 前記シフト手段は、2 段階のシフトを行うことを特徴とする請求項 7 記載の符号化装置。

【請求項 9】 前記シフト手段は、シフト量を上位ビットと下位ビットに分割してそれぞれシフトを行うことを特徴とする請求項 8 記載の符号化装置。

【請求項 10】 前記シフト手段はバレルシフタであることを特徴とする請求項 6 乃至 9 のいずれかに記載の符号化装置。

【請求項 11】 前記制御手段は、前記連結手段において形成した符号ビット列中に所定バイトを検出した場合に、該所定バイトの後に所定コードを挿入し、前記符号ビット列の前記所定バイト以降を再度連結することを特徴とする請求項 1 記載の符号化装置。

【請求項 12】 前記所定バイトの検出は AND ゲートにより行うことを特徴とする請求項 11 記載の符号化装置。

【請求項 13】 前記所定バイトの検出はレジスタにより 1 バイト毎に行うことを特徴とする請求項 11 記載の符号化装置。

【請求項 14】 前記所定バイトは“FF”バイトであることを特徴とする請求項 11 乃至 13 のいずれかに記載の符号化装置。

【請求項 15】 前記所定コードはビットスタフイングのためのコードであることを特徴とする請求項 1 または請求項 14 記載の符号化装置。

【請求項 16】 前記所定コードは制御コードであることを特徴とする請求項 1 記載の符号化装置。

10 【請求項 17】 前記所定コードはフィルビットのためのコードであることを特徴とする請求項 1 記載の符号化装置。

【請求項 18】 前記符号化手段はハフマン符号化を行うことを特徴とする請求項 1 記載の符号化装置。

【請求項 19】 所定の画素ブロック単位に可変長符号化を行う符号化装置であって、

前記画素ブロック内の所定単位毎に可変長符号化を行う符号化手段と、

前記符号化手段によって生成された符号データを連結して符号ビット列を形成する連結手段と、

20 前記符号ビット列に第 1 のコードを挿入する挿入手段と、

前記連結手段と前記挿入手段とを制御する制御手段とを有し、

前記制御手段は、前記連結手段において形成した符号ビット列中に所定バイトを検出した場合に、該所定バイトの後に第 2 のコードを挿入し、前記符号ビット列の前記所定バイト以降を再度連結することを特徴とする符号化装置。

30 【請求項 20】 前記第 1 のコードは制御コードであり、前記第 2 のコードはビットスタフイング又はフィルビットのためのコードであることを特徴とする請求項 19 記載の符号化装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は符号化装置に関し、例えば、可変長符号化の際に特殊コードの挿入を行う符号化装置に関する。

【0002】

【従来の技術】近年の通信装置や情報処理装置の発達に伴い、より効率的なデータ通信や、より効率的なデータ格納を行うために、データの符号化は不可欠の技術となり、種々の符号化方法が提案されている。

【0003】符号化方法は、符号が固定長である固定長符号化方式と、符号が必ずしも固定長でない可変長符号化方式とに大きく分けられる。

【0004】可変長符号化方式においては、符号中に、制御コードとして特殊コードを挿入する方法がしばしば用いられる。

50 【0005】通常、例えばデータ通信を行う際に符号中にノイズが混入し、ビットの反転等が発生することによ

り一旦符号が乱れてしまうと、それ以降の符号の復号が不可能となってしまう。しかしながら、特殊コードを挿入する符号化方式によれば、符号中の特殊コードを発見することにより、該特殊コード以降の符号から再び復号することが可能となる。

【0006】また、特殊コードに復号条件の指示機能を持たせることにより、復号装置側において、該特殊コード以降の符号の復号方法を変更することもできる。

【0007】このように、特殊コードを挿入する可変長符号化方式においては、柔軟な符号化が可能である。

【0008】可変長符号のビットストリーム（ビット列）の中に特殊コードを挿入するためには、通常のビットストリームの生成とは手順が異なる。例えば、一旦ビットストリームを生成した後にソフト的に特殊コードを挿入したり、又はビットストリームの生成過程（符号化プロセス）において、特殊コード挿入の都度符号化を止めて、特殊コードをソフト的な手段で挿入する等の方法により、特殊コードの挿入を実現していた。

【0009】

【発明が解決しようとする課題】しかしながら、上記従来の方法による特殊コードの挿入は、ビットストリーム内に特殊コードをソフト的な手段で挿入するために、ビットストリームを常にモニタしておく必要があり、従って符号化を司るCPUにかなりの負荷がかかってしまっていた。また、ソフト的な挿入を行うため、ビットストリームの生成速度が制限されてしまい、高速処理に限界が生じてしまっていた。

【0010】もちろん、該特殊コードの挿入をハード的に実現できれば、より高速な符号化が可能となるが、従来の符号化を行う装置においては、簡単な回路構成によってハード的に特殊コードの挿入を実現するための具体的な手段がなかった。

【0011】また、ビット列をつなげるために使用するバレルシフタは回路規模が大きくなってしまい、ハード化の妨げになっていた。

【0012】本発明は上述した課題を解決するためになされたものであり、小規模な回路構成によってビットストリームへの特殊コードのハード的な挿入を可能とする符号化装置を提供することを目的とする。

【0013】

【課題を解決するための手段】上記目的を達成するための一手段として、本発明の符号化装置は以下の構成を備える。

【0014】即ち、所定の画素ブロック単位に可変長符号化を行う符号化装置であって、前記画素ブロック内の所定単位毎に可変長符号化を行う符号化手段と、前記符号化手段によって生成された符号データを連結して符号ビット列を形成する連結手段と、前記符号ビット列に所定コードを挿入する挿入手段と、前記連結手段と前記挿入手段とを制御する制御手段とを有することを特徴とす

る。

【0015】例えば、前記所定コードは前記制御手段において生成され、前記挿入手段は、前記制御手段の制御に応じて前記符号化手段の出力と前記所定コードのいずれかを選択して前記連結手段に出力することを特徴とする。

【0016】例えば、前記連結手段は、入力されたデータを一時保持する保持手段を複数備えることを特徴とする。

【0017】例えば、前記連結手段は、前記保持手段を2つ有することを特徴とする。

【0018】例えば、前記連結手段は、ダブルバッファを有することを特徴とする。

【0019】更に、前記挿入手段により出力されたデータを前記連結手段における連結状況を考慮してシフトするシフト手段を有し、前記連結手段は、前記シフト手段によるシフト出力を入力することを特徴とする。

【0020】例えば、前記シフト手段は、前記保持手段の空き状況に応じたシフトを行うことを特徴とする。

【0021】例えば、前記シフト手段は、2段階のシフトを行うことを特徴とする。

【0022】例えば、前記シフト手段は、シフト量を上位ビットと下位ビットに分割してそれぞれシフトを行うことを特徴とする。

【0023】例えば、前記シフト手段はバレルシフタであることを特徴とする。

【0024】例えば、前記制御手段は、前記連結手段において形成した符号ビット列中に所定バイトを検出した場合に、該所定バイトの後に所定コードを挿入し、前記符号ビット列の前記所定バイト以降を再度連結することを特徴とする。

【0025】例えば、前記所定バイトの検出はANDゲートにより行うことを特徴とする。

【0026】例えば、前記所定バイトの検出はレジスタにより1バイト毎に行うことを特徴とする。

【0027】例えば、前記所定バイトは“FF”バイトであることを特徴とする。

【0028】例えば、前記所定コードはビットスタフイングのためのコードであることを特徴とする。

【0029】例えば、前記所定コードは制御コードであることを特徴とする。

【0030】例えば、前記所定コードはフィルビットのためのコードであることを特徴とする。

【0031】例えば、前記符号化手段はハフマン符号化を行うことを特徴とする。

【0032】また、所定の画素ブロック単位に可変長符号化を行う符号化装置であって、前記画素ブロック内の所定単位毎に可変長符号化を行う符号化手段と、前記符号化手段によって生成された符号データを連結して符号ビット列を形成する連結手段と、前記符号ビット列に第

1 のコードを挿入する挿入手段と、前記連結手段と前記挿入手段とを制御する制御手段とを有し、前記制御手段は、前記連結手段において形成した符号ビット列中に所定バイトを検出した場合に、該所定バイトの後に第 2 のコードを挿入し、前記符号ビット列の前記所定バイト以降を再度連結することを特徴とする。

【0033】例えば、前記第 1 のコードは制御コードであり、前記第 2 のコードはビットスタフニング又はフィルビットのためのコードであることを特徴とする。

【0034】以上の構成により、特殊コードの挿入と可変長符号の付加とを共通の構成により実現し、ハードウェアの負荷を大幅に軽減し、かつ高速に特殊コードの挿入が可能になった。

【0035】また更に、バレルシフタにおけるシフト量を 2 段階に分けることにより、回路規模が軽減された。

【0036】

【発明の実施の形態】以下、本発明に係る一実施形態について、図面を参照して詳細に説明する。

【0037】＜第 1 実施形態＞まず、本実施形態において扱う符号の特徴について説明する。

【0038】本実施形態で扱う可変長符号は、多値画像の圧縮を行う J P E G 標準圧縮の中の一部を構成するハフマン符号である。本実施形態での符号化においては、ハフマン符号とハフマン符号の間に付加ビットを挿入する操作が発生する。

【0039】ここで J P E G 符号化について簡単に説明する。まず、画素ブロック毎に多値データを離散的コサイン変換 (D C T 変換) し、量子化を施すと、量子化後の係数が “0” になるものが多く発生する。そして、各ブロック毎にジグザグスキャンにより画素を整理させ、0 ラン長とそれに続く 0 以外の係数の幅 (これをカテゴリと呼ぶ) をセットにしてハフマン符号を割り当てる。そして、該ハフマン符号の後に 0 以外の係数 (付加ビットと称する) を続ける。但し、J P E G 符号においてハフマン符号と付加ビットとは常に交互に出現する訳ではなく、例えばブロックの終端等、付加ビットが存在しないケースもある。

【0040】また、ハフマンコード中に出現する特殊コードとしては、例えばリスタートコード等、複数存在する。このリスタートコードは、通信エラー等でハフマンコードに間違いが発生した場合に、デコードを復帰させるために参照される。

【0041】これら特殊コードの挿入を行うためには、まず、生成されたハフマンコードのビット列に対し、バイトの区切りまでビット “1” を詰める。以降、これをフィルバイト操作と称する。そして、その後に “F F” バイトを挿入した後、“0 0” バイト以外からなる特殊コードを挿入する。即ち、“F F” バイトは、後に特殊コードが続くことを示す認識バイトである。

【0042】従って、ハフマンコードとその付加ビット

からなるビットストリーム中に、もし “F F” バイトが偶然に発生した場合や、フィルバイト操作の際に “F F” バイトが発生した場合には、発生した “F F” バイトの直後に意識的に “0 0” バイトを挿入することにより、特殊コードとの区別を可能としなければならない。以降、この操作をビットスタフニングと称する。

【0043】本実施形態の符号化装置においては、簡単な回路構成で上述したフィルバイト操作、特殊コード挿入、ビットスタフニングを可能としたことを特徴とする。

【0044】図 1 に、本実施形態における符号化装置のブロック構成を示す。本実施形態の符号化装置は、ハフマンエンコーダ及び特殊コード (以降、マーカコードと称する) の挿入回路構成を含むことを特徴とする。

【0045】図 1 において、R A M 1-1 ~ 1-4 はハフマンコードを発生 (生成) するためのエンコードテーブルであって、J P E G 標準圧縮で規定された成分の圧縮に対応し、輝度成分と色差成分、及びその 2 つの成分の D C T 結果に対する D C 成分と A C 成分用の 4 つのテーブルから成る。そして、セレクトラコントロール部 2 が現在の処理成分を判断して、R A M 1-1 ~ 1-4 のうち必要なテーブルの出力を選択するように、セレクトラ 3 をコントロールする。

【0046】図 2 に、エンコードテーブルである R A M 1-1 ~ 1-4 のワード構成を示す。本実施形態のエンコードテーブルは、最長 16 ビットのコードに対応するために 16 ビットのハフマン符号領域及び符号長 4 ビットの領域から成っており、ハフマン符号領域には右詰めでハフマンコードが格納されている。これらエンコードテーブルよりハフマンコードを得る際の R A M 1-1 ~ 1-4 へのアドレッシングは、入力されるバックデータから得られるアドレスにより行われる。

【0047】ここで図 3 に、本実施形態の入力であるバックデータの構成例を示す。本実施形態のバックデータは、上述した J P E G 符号化に基づいて画素ブロックに対し D C T 変換、および量子化を施した後の、D C 係数や A C 係数の並び (ジグザグスキャン順) を示すものである。このバックデータにおける D C カテゴリや A C カテゴリ、及び 0 ラン等のビット列が、そのままエンコードテーブルに対するアドレス情報となり得る。また、図 3 に示す D C 差分や A C 成分は前述した付加ビットに相当し、R A M 1-1 ~ 1-4 のアクセスには使用されず、図 1 に示す合成部 6 に送られる。

【0048】図 3 に示すように、バックデータには 2 ビットからなる識別フラグ領域が存在し、該識別フラグは、セレクトラコントロール部 2 において D C 成分や A C 成分の判別の際に使用されたり、また、付加ビットであるか否かの判定等に使用される。尚、識別フラグ “1 1” で示される E O B は、ブロックの終端まで 0 ランが続く場合を示す。

【0049】図1に戻り、セクタ3より得られたハフマン符号は、シフト5によって左方向(MSB方向)にシフトされ、合成部6でLSB側に付加ビットデータが付される。シフト5におけるシフト量は、パックデータ内のDCカテゴリ又はACカテゴリにより定まる。尚、ハフマンコードの次に付加ビットを付す必要がない場合には、シフト5におけるシフト量は“0”となる。

【0050】また、加算部4において、セクタ3より得られたハフマン符号に対応する符号長がパックデータ内のDCカテゴリ又はACカテゴリにより示される付加データ長と加算され、合計の符号長が求められる。そして、該符号長は特殊コード制御部10へ入力される。

【0051】ところで、ハフマンコードと付加ビットとが合成される合成部6においては、ハフマンコードより上位のビットをマスクして“0”にする操作を施しても良いが、本実施形態においては該操作を行わずとも、後述する合成用レジスタ部9においてハフマンコードよりも上位のビットは無視される。

【0052】合成部6の出力は、通常はセクタ7を介してパレルシフト8へ与えられるが、本実施形態ではセクタ7に特殊コード制御部10より特殊コードが与えられており、セクタ7においては合成部6の出力と該特殊コードのいずれかを選択してパレルシフト8に与える。このセクタ7による選択出力により、本実施形態においてハフマンコード列の間に特殊コードを挿入することが可能になる。

【0053】パレルシフト8では、合成用レジスタ部9における空き領域に連続したビット列を形成するために、入力されたデータに対して適当なビット数のシフトを行った後、合成用レジスタ部9に出力する。合成用レジスタ部9は2つのレジスタを有しており、片方のレジスタが一杯になると、該レジスタ内に形成されたビット列をはき出し、空きのあるレジスタ、又は空きのあるレジスタと未使用のレジスタにパレルシフト8からの入力を保持する機能を有する。

【0054】以下、図4を参照して、合成用レジスタ部9における動作を更に詳細に説明する。

【0055】図4の(a)～(c)は、シフト5に入力され、合成部6から出力されるハフマン符号データの構造を模式的に示す図である。まず、シフト5に入力されるハフマン符号データは図4の(a)の斜線で示され、ハフマン符号の符号長を超える部分には“0”が詰められている。この“0”はエンコーダRAM1-1等に保持されていても良いし、また、“0”以外の値であっても問題は生じない。

【0056】図4の(b)は、シフト5において入力されたハフマン符号データを、LSBに付加する付加ビット長分だけMSB方向へシフトした状態を示す。図4の(c)は、合成部6においてハフマン符号データと付加ビットとが連続するビット列になる様に合成された状態

を示す。ここで、シフト5においてハフマン符号データをMSB方向へシフトした際に空いたLSB側に“0”を詰め、かつパックデータの付加ビットのビット長を超える上位ビットを“0”とすれば、合成部6は2つの入力の論理和をとるORゲートによって簡単に構成できる。

【0057】このようにして、合成部6においては図4の(c)で示される様に合計bビットのデータが作られるとする。尚、合成部6としてはセクタを使用し、LSB側は付加ビット、MSB側はシフト5の出力を選択するようにし、付加ビット長に応じてLSB側、MSB側のいずれかの選択を行うように構成しても良い。

【0058】図4の(e)、(f)は、合成用レジスタ部9内の2本のレジスタ内容をそれぞれ模式的に示す図である。ここでは、各レジスタは32ビット幅であるとする。

【0059】例えば、図4の(f)に示す片方のレジスタが空で、図4の(e)に示す様にもう片方のレジスタにはaビット長のデータが既に詰まっている場合には、パレルシフト8において図4の(c)に示されるハフマン符号データがシフトされる際のシフト量は、 $32 - (a + b)$ ビットとなる。このシフト量 $32 - (a + b)$ が負となれば、シフト方向が反対、又は負の値に32を加えたシフトを行うという意味である。

【0060】尚、図4の(d)はパレルシフト8からのシフト後の出力データ例を示し、合成用レジスタ部9内のデータと同様、32ビットである。図4の(d)においては、 $a + b$ が32ビットを越えた場合、即ち、パレルシフト8におけるシフト量が負である場合のシフト例を示している。

【0061】このパレルシフト8の出力は、合成用レジスタ部9内の図4の(e)に示すレジスタにおいて、 $d = 32 - a$ ビットのLSB部分のみがラッチされる。そして、もう一方の図4の(f)に示すレジスタにおいては、 $b - d$ ビット分がMSB部分にラッチされる。これにより、図4の(e)に示すレジスタは満杯となるため、合成用レジスタ部9から外部に出力されるデータとして処理される。

【0062】このようにして、合成用レジスタ部9においては、片方のレジスタが一杯になると出力して空にする。そして、同様の動作を繰り返すことにより、本実施形態において連続したビット列を形成することが可能となる。従って、合成用レジスタ部9における片方のレジスタは常に空の状態でスタンバイしていることになる。

【0063】ここで、 $a + b$ が32ビットを超えない場合、即ち、パレルシフト8におけるシフト量が正である場合の、パレルシフト8の出力例を図4の(g)に示す。また、図4の(h)は、パレルシフト8の出力のうち、有効なbビットが合成用レジスタ部9内の一方のレジスタに書き込まれた状態を示す。このような状態にお

10

20

30

40

50

いては、合成用レジスタ部 9 の他方のレジスタに対する書き込みは行われない。

【0064】また、特殊コード制御部 10 においては、特殊コードの挿入以外にパレルシフタ 8 のシフト量を合成用レジスタ部 9 の書き込み状態に基づいて判断したり、合成用レジスタ部 9 から 32 ビットにパックされた連続するビットストリームの出力の制御等を司る。

【0065】以下、以上説明した合成用レジスタ 9 の更に詳細な構成を図 5 に示し、説明する。尚、図 5 においては、特殊コード制御部 10 における構成が一部含まれており、合成用レジスタ 9 と特殊コード制御部 10 との境界を点線で表わしている。

【0066】図 5 において、レジスタ A 21 及びレジスタ B 22 が、上述した図 4 において示した (e)、(f) の 32 ビットレジスタにそれぞれ対応する。

【0067】例えばレジスタ A 21 に既に a ビットのコードが書き込まれているとし、次に書き込むコードのコード長を b ($b = \text{ハフマン符号長} + \text{付加ビット長}$) とすれば、パレルシフタ 8 におけるシフト量は、上述した様に $32 - (a + b)$ ビットとなり、パレルシフタ 8 の出力の一部がレジスタ A 21、B 22 にラッチされる。

【0068】より詳細には、特殊コード制御部 10 内に構成されるマスク生成部 23、24、及びラッチ制御部 25 によって、各レジスタに対するマスクデータが生成され、該マスクデータにより、上述した様にレジスタ A 21 では既に書き込まれている上位 a ビットを除いたデータがラッチされ、レジスタ B 22 では上位の $(a + b) - 32$ 、又は $(b - d)$ のビット数がラッチされる。

【0069】ここで、 $(a + b) - 32$ が負の場合はレジスタ B 22 にはラッチされない。一方、正の場合にはレジスタ B 22 にラッチが行なわれると同時に、レジスタ A 21 がコードで一杯になるため、レジスタ A 21 の 32 ビットデータは後段の回路にはき出され空となる。これにより、レジスタ A 21 と B 22 とは役割が交代する。

【0070】ここで、マスク生成について説明する。特殊コード制御部 10 内のマスク生成部 23、24 においては、値 a 及び値 $(a + b) - 32$ に基づいてレジスタ A 21 又はレジスタ B 22 への書き込み位置を判定し、マスクデータを生成する。該マスクデータは、ラッチ制御部 25 を介してレジスタ A 21、レジスタ B 22 へ与えられるが、上述した様にレジスタ A 21 とレジスタ B 22 の役割が交代すると、ラッチ制御部 25 における 2 つの出力も入れ換えが行われる。

【0071】レジスタ A 21、B 22 から出力される各 32 ビット、即ち合計 8 バイトは、AND ゲート 28 において各バイト毎の AND がとられ、それぞれ“FF”のバイトであるか否かが判定される。この時、既に処理が終了している a ビットを処理対象から除外するため

に、マスク生成部 30 によりゲート 34 を介して a ビット分のマスクがかけられる。

【0072】又、“FF”を伴うマーカ（特殊コード）も、図 1 に示した特殊コード制御部 10 よりセクタ 7 を介してレジスタ A 21、B 22 に与えられるため、マーカにおける“FF”の位置に応じて、ゲート 34 を介してマスク生成部 32 により生成されたマスクがかけられる。

【0073】更に、処理済みの a ビットと処理中のハフマン符号 b ビットの和である $(a + b)$ ビットを除く残りの部分についても、不要な情報が格納されているおそれがあるため、ゲート 34 を介してマスク生成部 31 により生成されたマスクをかける。

【0074】ここで、“FF”バイトが発見できなかった場合には $a_{\text{new}} = a + b$ となるが、“FF”バイトが発見された場合には、プライオリティエンコード 35 によりその位置 f を発見し、 $a_{\text{new}} = f$ とする。即ち、該“FF”の位置までを処理済みとする。

【0075】特殊コードであるマーカは、“FF”バイト及びそれに続く“0”以外のバイトから成り、マーカ挿入位置に応じて補正部 33 が出力する情報に従ってマスク生成部 32 により生成されたマスクがゲート 34 を介してかけられる。

【0076】以上説明した様に、“FF”バイトが発見された場合には、実際の処理済みビット数 a は更新され、 $a_{\text{new}} = f$ となる。この後、特殊コード制御部 10 では“00”をセクタ 7 へ出力することによって“00”バイトの挿入、即ちビットスタッフィングを行う。そしてその後に、残りビット分 (c ビットとする) をパレルシフタ 8 を介して再度レジスタ A 21、B 22 に対してラッチする必要がある。

【0077】ここで図 6 に示すように、b ビットの内、ビットスタッフィング前までのビット列を書き込み終った残り部分が c ビットとなり、これは、パレルシフタ 8 の入力部に保持され続ける。ここで c ビットは前回の処理数 b ビットに対して、以下の式で表される。

$$\text{【0078】} \quad c = b_{\text{new}} = b - (f - a) \quad \dots (1)$$

尚、式 (1) において $(f - a)$ は前回の処理済みビット数である。

【0079】従ってパレルシフタ 8 では、この c ビットを処理するために、

$$32 - (a' + b_{\text{new}}) \quad \dots (2)$$

のシフトが必要となる。ここで、

$$a' = a_{\text{new}} + 8 \quad \dots (3)$$

$a_{\text{new}} = f$ 、8 = スタッフィングビット数

であるから、(2) 式に対して (1)、(3) 式より

a' 、 b_{new} を代入することにより、(2) 式は $32 -$

$(a + b + 8)$ となる。即ち、パレルシフタ 8 におけるシフト量は、ビットスタッフィングのために 8 ビットず

れることになる。

【0080】尚、前回の処理済みビット数 $f-a$ は、減算器 36 より得られる。そして、通常は $(a+b)$ が 32 以上、又、ビットスタッキング発生時は f が 32 以上になると、セクタ 26 を介してレジスタ A 21、B 22 の一杯となった値 (32 ビット) が出力され、レジスタ 27 を介して圧縮データとして外部に出力される。

【0081】また、フィルバイト操作時 (マーカコード挿入直前) には、 a と、 a より大きい 8 の倍数との差分を b とし、セクタ 7 及びバレルシフタ 8 を介して、合成用レジスタ 9 に対してビット “1” の書き込みを行えば良い。

【0082】尚、合成用レジスタ 9 と特殊コード制御部 10 との機能の振り分けは、必ずしも図 5 に示す通りである必要はなく、例えば AND ゲート 28 等も特殊コード制御部 10 に含まれるような構成であっても良い。即ち、合成用レジスタ 9 と特殊コード制御部 10 とによって、図 5 に示す構成を満たせば良い。

【0083】次に、バレルシフタ 8 の詳細構成を図 7 に示し、以下説明する。

【0084】本実施形態のバレルシフタ 8 は、32 ビット幅の出力を有する 32 ビットバレルシフタであるが、説明の簡便のため、図 7 に 8 ビットバレルシフタの構成例を示し、説明する。

【0085】図 7 において、81 は入力を受けるバッファであり、D0〜D7 までの信号を後段に伝える。この信号は 8 to 1 セクタ 82-1〜82-8 に、それぞれ 1 ビットずつ順番をずらして入力されている。従って、セクタ 82-1〜82-8 の選択制御信号として同一の値を与えれば、1 ビットずつずれた値が S0〜S7 に得られる。従って、セクタ 82-1〜82-8 に対する選択制御信号として、希望するシフト量を与えれば良い。

【0086】従って、本実施形態の 32 ビットバレルシフタ 8 は、図 7 におけるバッファ 81 及びセクタ 82-1〜82-8 を、32 ビット対応として実現すれば良い。

【0087】以上説明した様に本実施形態によれば、特殊コードの挿入とそれに関連する操作を可変長符号の付加を行う合成用回路やバレルシフタ等を共有して実現することにより、ハードウェアの負荷を大幅に軽減し、小規模な回路構成で特殊コードの高速挿入が可能となった。

【0088】＜第 2 実施形態＞以下、本発明に係る第 2 実施形態について説明する。第 2 実施形態においては、上述した第 1 実施形態とは合成用レジスタ 9 の詳細構成が異なる。

【0089】図 8 に、第 2 実施形態における合成用レジスタ 9 の詳細構成を示す。図 8 において、上述した第 1 実施形態で示した図 5 と同様の構成については同一番号

を付し、説明を省略する。

【0090】図 8 に示す構成においては、基本的には、レジスタ A 21、又はレジスタ B 22 が一杯になると、セクタ 26 を介して後段の回路へデータが送られる。この仕組みは上述した第 1 実施形態で説明した通りである。

【0091】第 2 実施形態においては、セクタ 26 の後段回路として 2 組の 4 バイトレジスタ 41、43 を有しており、R/W 制御部 44 及びセクタ 42 の制御により、片方のレジスタ 41 から 1 バイトずつ読みとって、他方のレジスタ 43 に “FF” か否かをチェックしながら書き込む。書き込み用のレジスタ 43 は、書き込み 4 バイト毎に、不図示の FIFO 又は出力用バッファへデータを出力する。

【0092】レジスタ 41 より読み取られた 1 バイトずつのデータは、ラッチ 45 を介して “FF” か否かがゲート 47 で判定される。“FF” であれば、書き込み用レジスタ 43 に “FF” を書き込んだ直後にゲート 46 を閉じ、書き込み用レジスタ 43 に “00” バイトを続けて書き込む。これにより即ち、第 2 実施形態におけるビットスタッキングが行われる。ただし、特殊コード制御部 10 において、自身が発生したマーカコード “FF” の位置に応じて、“FF” 発見の結果をキャンセルし、ビットスタッキングが発生しないように制御する必要がある。これは、マーカ発生部 50 及びその遅延部 49、及びゲート 48 により達成される。尚、第 2 実施形態におけるマーカコードの発生、及びフィルバイト操作による “1” の挿入は、上述した第 1 実施形態と同様の方法で実現される。

【0093】ちなみに第 2 実施形態におけるフィルバイト操作は、マーカ挿入の直前における処理済みビット数を a とすると、 $8 - (a \% 8)$ (但し、 $a \% 8$ は a を 8 で割った余り) のビット数だけ、全て “1” がセクタ 7 を介してバレルシフタ 8 の入力として与えられることにより、達成される。例えば、 $a = 9$ である場合、7 ビットが “1” としてバレルシフタ 8 に与えられる。もちろん、フィルバイト操作とマーカコード挿入とを同時にバレルシフタ 8 で行うことも可能である。

【0094】以上説明した様に第 2 実施形態によれば、第 1 実施形態とは異なる合成レジスタの構成によっても、特殊コードの挿入等を同様にを行うことができる。

【0095】＜第 3 実施形態＞以下、本発明に係る第 3 実施形態について説明する。第 3 実施形態においては、上述した第 1 実施形態とは合成用レジスタ 9 の詳細構成が異なる。

【0096】図 9 に、第 3 実施形態における合成用レジスタ 9 の詳細構成を示す。図 9 において、上述した第 2 実施形態で示した図 8 と同様の構成については同一番号を付し、説明を省略する。

【0097】上述した第 1 実施形態においては、バレル

シフタ 8 に対してフィルバイト “1” とマーカコードとを同時に与えた場合、フィルバイト操作の結果得られるバイトが “FF” であった場合、マーカコードを新たにセットし直さなくてはならない。

【0098】また、上述した第 2 実施形態においては、マーカコード挿入のためにフィルバイトを行った結果 “FF” となるバイトを考慮して、ビットスタフイングが適切に行われる様、注意して設計する必要がある。

【0099】第 3 実施形態においては、マーカコードの合成をフィルバイト操作及びビットスタフイングとは独立した構成によって実現することにより、係る不具合を解消したことを特徴とする。

【0100】図 9 に示す構成における基本動作は、上述した第 2 実施形態と同様である。従って、“00” を挿入するビットスタフイング、及びマーカコード挿入直前のフィルバイト “1” の挿入動作についても同様である。ただし、第 3 実施形態においては、マーカコードの挿入はセクタ 7、バレルシフタ 8 を介さず、セクタ 7 4 を介して、直接書き込み用レジスタ 4 3 に 1 バイトずつを順に書き込む。即ち、マーカコードの挿入をフィルバイト操作及びビットスタフイングとは独立した構成によって実現する第 3 実施形態においては、フィルバイト “1” を挿入したバイト境界後にマーカコードが入るように、マーカコードのタイミングを調整する必要がある。この調整は、フィルバイト管理部 7 5 及びタイミング調整のための遅延部 7 6 により、マーカ発生部 7 7 におけるタイミングを適当に合わせることで実現する。これにより、第 3 実施形態においてはマーカコード挿入時におけるバレルシフタ 8 への負荷を軽減することができる。

【0101】また、“00” を挿入するビットスタフイングの際に “FF” バイトを検出する必要があるが、この際にマーカコードの位置を考慮しなくて良いため、バイトレジスタ 7 2 を通過するデータに対してゲート 7 1 で “FF” か否かの判断を行う。そして、“FF” であればレジスタ 7 2 を 1 クロック分クリアして “00” バイトを挿入する。尚、レジスタ 7 3 はマーカコード等の書き出しレジスタとして使用される。

【0102】以上説明した様に第 3 実施形態によれば、マーカコードの合成をフィルバイト操作及びビットスタフイングとは独立した構成によって実現することにより、フィルバイトによって生じた “FF” バイトを特に考慮することなく、マーカコード挿入及びビットスタフイングが行える。

【0103】＜第 4 実施形態＞以下、本発明に係る第 4 実施形態について説明する。第 4 実施形態においては、32 ビットバレルシフタ 8 をより小さな規模の構成で実現する例について説明する。第 4 実施形態におけるバレルシフタ 8 の詳細構成を図 10 に示す。

【0104】まず、バッファ 8 3-1 ～ 8 3-4 は 8 ビ

ットバッファであり、それぞれ 8 ビットずつの入力を受ける。8 4-1 ～ 8 4-4 はセクタであり、該セクタにそれぞれ入力される 32 ビットは、8 ビット (1 バイト) の中の順番を変えずに 4 つのバイトをシフトして、順番がずらされる。即ち、8 ビット単位でのローテーションが行われる。

【0105】以下、セクタ 8 4-1 ～ 8 4-4 における動作について説明する。セクタ 8 4-1 ～ 8 4-4 は、それぞれ 4 to 1 セクタ 8 個からなる。各 4 to 1 セクタにはそれぞれバッファ 8 3-1 ～ 8 3-4 に保持された信号が、8 ビットずれた信号 (例えば D0, D8, D16, D24) として 4 つが入力され、そのうちの 1 つが選択される。そして、各セクタを構成する 8 個の 4 to 1 セクタには、それぞれ他の 4 to 1 セクタとは 1 ビットずつずれた信号が入力されることにより、同一の選択信号に基づいて、連続する 8 ビットの信号、即ち 1 バイトの信号 (例えば D0 ～ D8) が得られる。

【0106】又、各セクタ 8 4-1 ～ 8 4-4 には互いに 8 ビットずつずらした信号を入力することにより、出力される t0 ～ t31 にはバイト単位でのシフトが施されることになる。

【0107】ここで例えば、バレルシフタ 8 におけるシフト量を 5 ビット (0 ～ 31 のシフト量) とすると、まずセクタ 8 4-1 ～ 8 4-4 におけるバイト単位のシフト処理により、即ち上位 2 ビット (8 の整数倍) のシフトが行われたことになる。次に、下位 3 ビットのシフト (0 ～ 7 のシフト量) を行うが、そのためには 2 つのバイトにまたがったデータが必要となる。従って、入力される 16 ビットからシフト量に応じて 8 ビットを取り出すために、16 to 1 セクタを 8 個有する 16 ビットバレルシフタが 4 つ、即ちバレルシフタ 8 5-1 ～ 8 5-4 が必要となる。尚この場合、シフト量は 0 ～ 7 であるため、実際には 8 to 1 セクタを 8 個有するシフタが 4 つあれば済む。

【0108】8 to 1 セクタの各入力には、1 ビットずつずれた信号を 8 ビット分 (例えば t0, t1, … t8) 入力し、8 個の 8 to 1 セクタはそれぞれ他の 8 to 1 セクタと 1 ビットずれた信号が入力される。従って、バレルシフタ 8 5-1 ～ 8 5-4 の入力には 16 ビット幅が必要である。これは入力信号の違いを除けば、上述した第 1 実施形態において図 7 に示した 8 ビットバレルシフタと同様の構成で実現できる。

【0109】そして、バレルシフタ 8 5-1 ～ 8 5-4 にはそれぞれ 8 ビットずつずらした値を与えることにより、それぞれ連続した 8 ビットが 4 つのバレルシフタ 8 5-1 ～ 8 5-4 の出力同士で更に連続し、結果として 32 ビットバレルシフタ 8 の出力として得られる。これにより即ち、第 4 実施形態において全 5 ビット分のシフトが施されたことになる。

【0110】以上説明したように第 4 実施形態によれ

ば、図10に示す様にバレルシフトを2段構成にし、シフト量を上位ビットと下位ビットに分けることにより、第1実施形態において図7に示した8ビットバレルシフトの構成をそのまま32ビットバレルシフトとして拡張した場合よりも、回路規模が小さくて済む。

【0111】＜他の実施形態＞なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0112】

【発明の効果】以上説明した様に本発明によれば、特殊コードの挿入とそれに関連する操作を、可変長符号の付加と同一の合成用回路等を共有することで実現したことにより、ハードウェアへの負荷を大幅に軽減し、小規模の回路構成による高速な符号化処理が可能となる。

【0113】また、バレルシフトを2段構成にし、シフト量を上位ビット、下位ビットに分けてシフトをすることにより、更なる回路規模の低減が可能になった。

【0114】

【図面の簡単な説明】

【図1】本発明に係る一実施形態における符号化装置の構成を示すブロック図である。

【図2】本実施形態におけるエンコードRAMのワード構成図である。

【図3】本実施形態におけるパックデータ例を示す図である。

【図4】本実施形態における合成用レジスタの動作を説明するための図である。

【図5】本実施形態における合成用レジスタの詳細構成を示すブロック図である。

【図6】本実施形態における合成用レジスタの動作を説明するための図である。

【図7】本実施形態におけるバレルシフトの詳細構成を示す図である。

【図8】本発明に係る第2実施形態における合成用レジスタの詳細構成を示すブロック図である。

【図9】本発明に係る第3実施形態における合成用レジスタの詳細構成を示すブロック図である。

【図10】本発明に係る第4実施形態におけるバレルシフトの詳細構成を示すブロック図である。

【符号の説明】

1-1～1-4 エンコードRAM

2 セレクタコントロール部

3 セレクタ

4 加算器

5 シフト

6 合成部

7 セレクタ

8 バレルシフト

9 合成用レジスタ部

10 10 特殊コード制御部

21, 22 レジスタ

23, 24 マスク生成部

25 ラッチ制御部

26 セレクタ

27 レジスタ

28 ANDゲート

29 セレクタ

30, 31, 32 マスク生成部

33 補正部

20 34 ゲート

35 プライオリティエンコーダ

36 減算器

41, 43 レジスタ

42 セレクタ

44 R/W制御部

45 ラッチ

46 ゲート

47 ANDゲート

48 ゲート

30 49 遅延部

50 マーカ発生部

71 AND回路

72, 73 レジスタ

74 セレクタ

75 フィルバイト管理部

76 遅延部

77 マーカ発生部

81 入力部

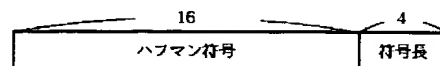
82-1～82-8 セレクタ

83 入力部

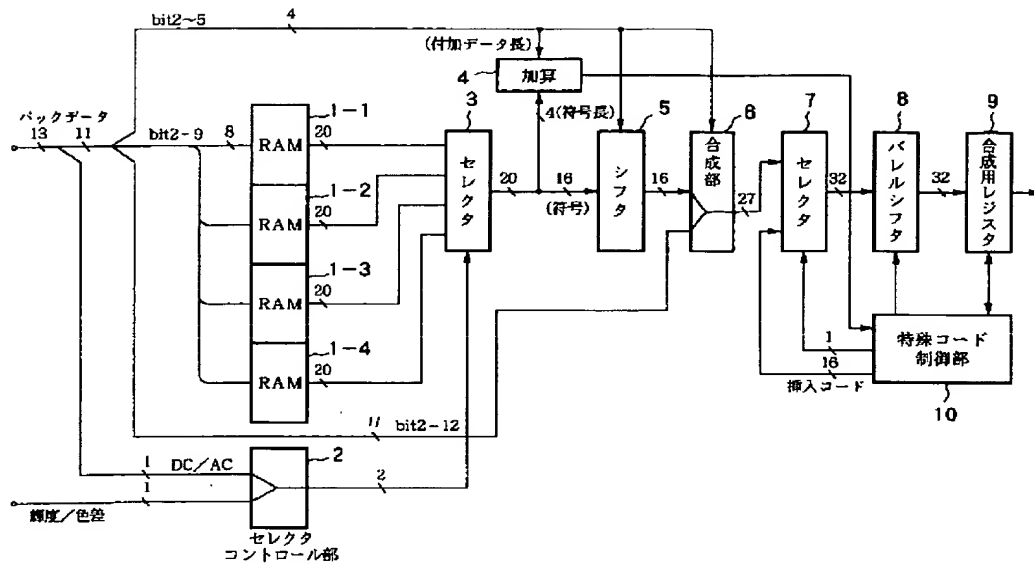
84-1～84-4 セレクタ

85-1～85-4 セレクタ

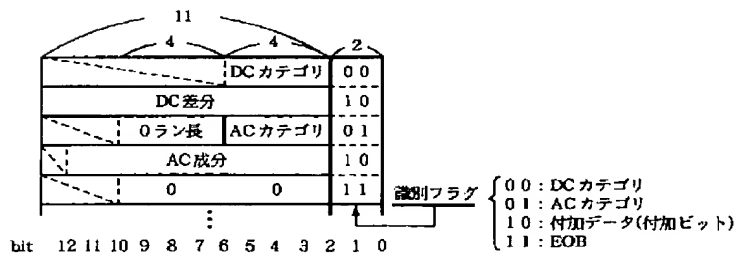
【図2】



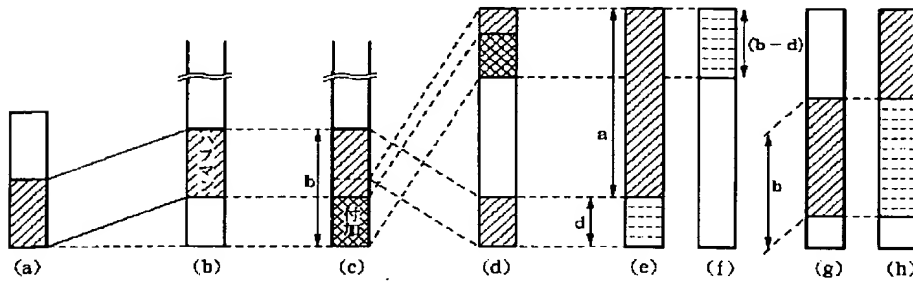
【図 1】



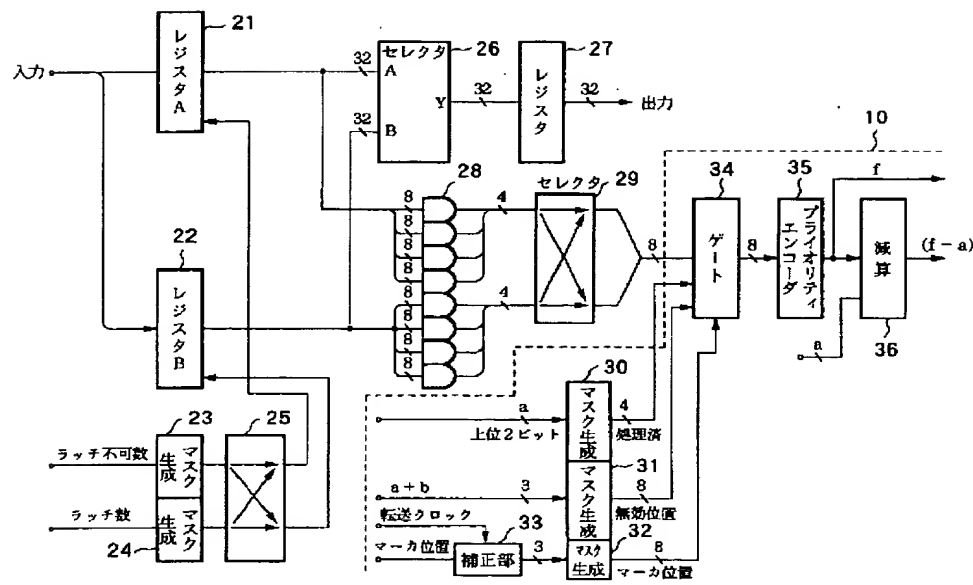
【図 3】



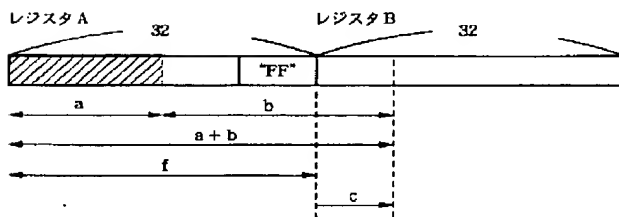
【図 4】



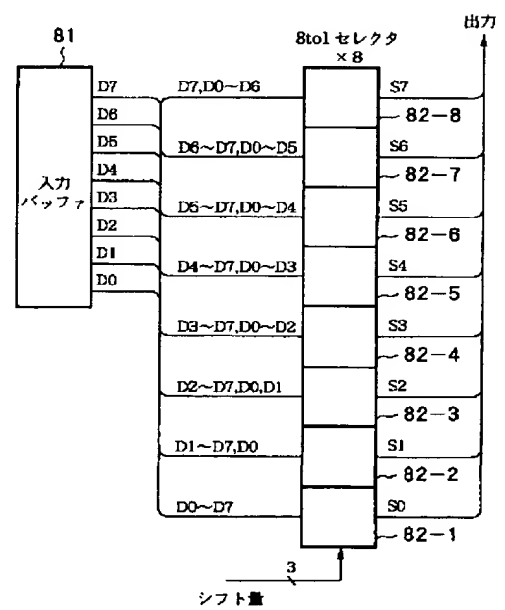
【図 5】



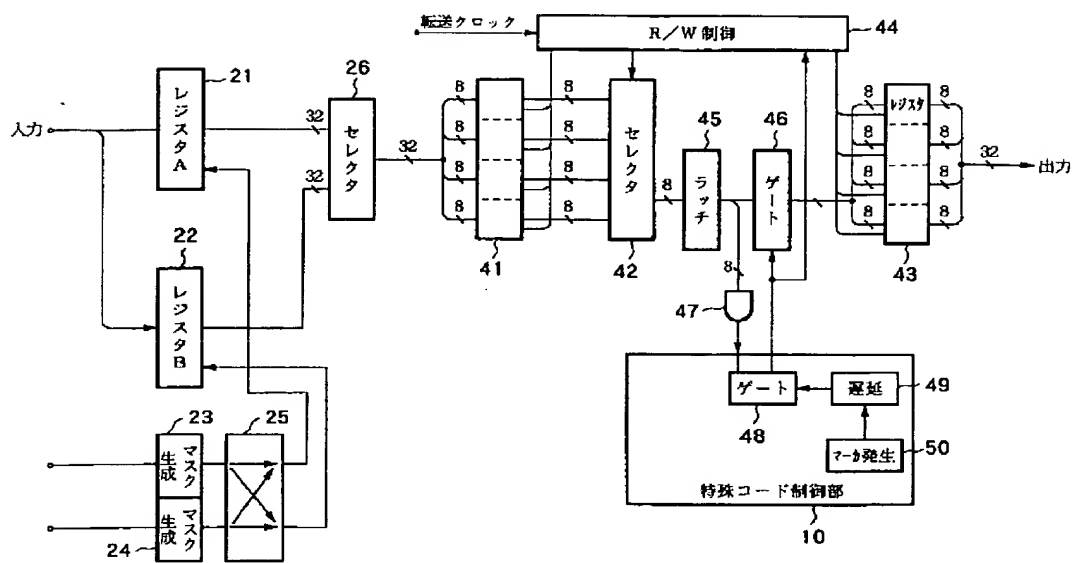
【図 6】



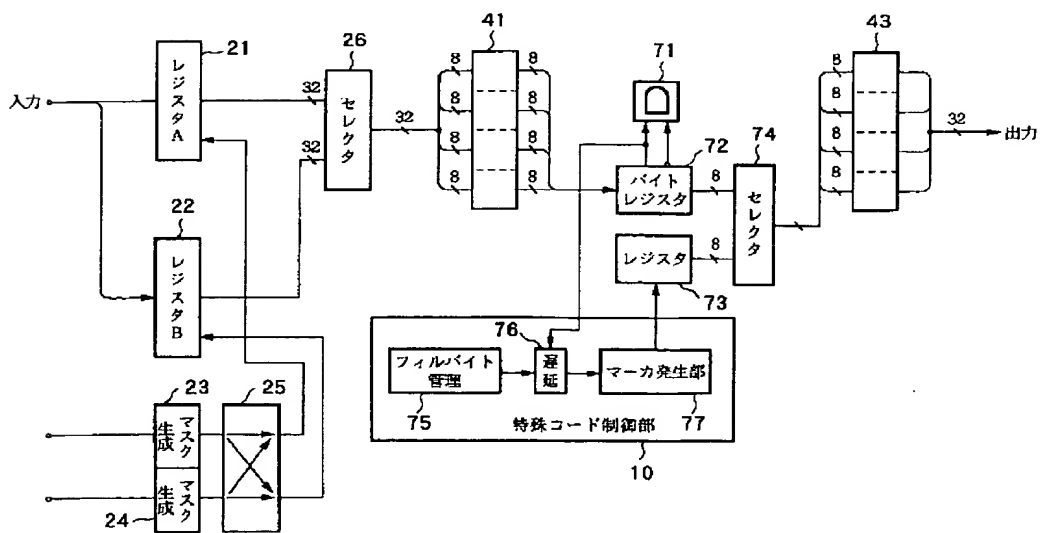
【図 7】



【図 8】



【図 9】



【図 10】

